

# Fixture knowledge model development and implementation based on a functional design approach

R. Hunter Alarcón<sup>a</sup>, J. Ríos Chueco<sup>b,\*</sup>, J.M. Pérez García<sup>b</sup>, A. Vizán Idoipe<sup>b</sup>

<sup>a</sup> Mechanical Engineering Department, Universidad de La Frontera, Av. Francisco Salazar 01145, Casilla 54-D, Temuco, Chile

<sup>b</sup> Mechanical and Manufacturing Engineering Department, Universidad Politécnica de Madrid, Jose Gutiérrez Abascal 2, 28006, Madrid, Spain

## ABSTRACT

The development of a knowledge model applied to fixture design is a complex task. The main purpose of such model is the development of a knowledge-based application to assist fixture designers. It comprises a detailed specification of the types and structures of data involved in the execution of the inference process needed to create a fixture solution for machining a raw part. A development method together with a knowledge model for automating fixture design is proposed. The development was divided into three parts: Design Process Model, definition of Top-level functional functions and Product Knowledge Model. Adopting a functional design approach, the fixture design solution was created in two levels: functional and detailed. The functional level is based on fixture functional elements and the detailed one is based on fixture commercial elements. The definitions and concepts used in the application are specified in several Units of Knowledge (UoK) that comprises the Fixture Knowledge Model. Common Knowledge Analysis and Design Structuring (CommonKADS), Methodology and software tools Oriented to KBE Applications (MOKA), Integrated DEfinition for Function Modelling (IDEFO) and Unified Modelling Language (UML) are the methodologies and techniques used in the proposed method. Finally, a prototype KBE application for fixture design was developed.

## Keywords:

Knowledge model  
KBE development  
Fixture design  
Fixture KBE application

## 1. Introduction

Knowledge-based systems (KBS) have evolved from computer programs that partially automate the creation of specific solutions for design problems, formerly known as expert systems, to the development of integrated systems that support designers' decisions along the product lifecycle. In this evolution, there are key elements such as the definition of an integrated design process, the identification of knowledge intensive tasks, the capture of experts' knowledge and the modelling of all of it. For that purpose the use of methodologies such as Common Knowledge Analysis and Design Structuring (CommonKADS) [1] and Methodology and software tools Oriented to KBE Applications (MOKA) [2] is highly recommended.

Knowledge-based engineering (KBE) applications are a specific type of KBS. They are tightly linked to product supporting lifecycle packages such as computer-aided design (CAD) applications. In their development there is a need to capture from experts the 'what' of the design, the 'how' or the steps to carry out and also the 'why'. In consequence, to automate the fixture design process two

main actions have to be undertaken: the modelling of the design process and the modelling of the knowledge needed [2].

The design of fixtures demands extensive use of heuristic knowledge, which is also coupled. For instance, when defining the machining operations, the fixture solution should be kept in mind and vice versa. Experience and skills, gathered and kept by the designers in the form of explicit and tacit knowledge for several years, are essential in achieving proper fixture designs. Two basic factors have to be also considered:

1. The extensive information needed during the design process, mainly related to part definition, machining processes, manufacturing resources, fixture elements and production requirements.
2. The complexity of the design itself that implies the determination of: locating, supporting and clamping positions and the corresponding physical fixture elements, whereas satisfying requirements and constraints regarding: stability, rigidity, deformability, accuracy, accessibility, interference, availability and cost.

This makes very difficult to completely automate the design of fixtures within an industrial environment. With the improvements in feature-based design, geometry analysis algorithms, information exchange, knowledge capture and representation and

artificial intelligence, the development of such applications has been facilitated, but still the extensive expertise needed during the process makes this area of research highly challenging [3–8].

This paper presents a fixture design process and the knowledge models used to develop a KBE application for modular fixture design. The Integrated DEfinition for function modelling (IDEFO) technique was used to define the fixture design process and to identify Units of Knowledge (UoK). From the MOKA methodology, the forms named: illustrations, constraints, activities, rules and entities (ICARE); were employed to capture and represent experts' knowledge. They were used to document elements of the type: input, control, output and mechanism (ICOM); defined for each activity in the IDEFO model. A knowledge template methodology from CommonKADS and Unified Modelling Language (UML) was then used to create the domain knowledge. Finally, such model was implemented in a KBE prototype application using the development environment of a commercial system.

## 2. Methodologies to develop knowledge-based applications

CommonKADS is a general methodology focused on knowledge management and knowledge-based systems. The development of such systems has to be done independently of any possible implementation technique. It requires specifying the knowledge and the reasoning requirements. Such specification is part of the analysis process and contains the data and knowledge structures needed for the application. It proposes three modelling steps: context modelling, knowledge modelling and communication modelling. The knowledge model is divided into three parts: domain knowledge, inference knowledge and task knowledge. The domain knowledge contains all the definitions of the concepts used in the application. It is a static view and it corresponds to a data model or information model as it is called in software engineering. The task knowledge defines the objectives to be obtained and how to achieve them by using inferences [1]. MOKA has its origins directly linked to CAD systems and it is influenced by CommonKADS amongst other techniques. Knowledge-based engineering pursues to capture and reuse product and process knowledge [2].

Any KBE system in the design environment needs a link with a geometric modelling kernel. Parasolid, ACIS and Open Cascade are some of the geometric kernels used by some KBE systems. Development environments for KBE provide an inference engine that derives answers from the knowledge base, e.g. Intelligent-CAD (ICAD). When that is not the case, then an inference knowledge model has to be created to develop the application. That is the general case considered in CommonKADS. On the other hand, MOKA considers the availability of such inference engine. MOKA defines two models to be created: Design Process Model and Product Model. The Design Process Model is defined as activity diagrams. For the Product Model, MOKA provides two levels of knowledge representation: informal level and formal level. The informal level is based on the called ICARE forms. The formal level requires transforming the knowledge defined in such forms into MOKA Modelling Language (UML-based diagrams) [2].

## 3. Approach to develop a KBE application for design of fixtures

Both MOKA and CommonKADS constitute the core of the method followed in this research, together with two modelling techniques: IDEFO and UML. It contains the following main outputs: Design Process Model, Definition of Top-Level Functions, Product Knowledge Model, mapping into implementation system data structure and Application Program Interface (API), models

specific to the implementation system and fixture KBE application coding (Fig. 1).

The Fixture Design Process Model was represented using IDEFO activity diagrams [9]. Such model represents the tasks to carry out, the flow of information and the UoK to be modelled in the Product Knowledge Model. Since the design of fixtures is integrated with product design and manufacturing, complementary to the IDEFO model, a scenario of use for the application was defined. This scenario was documented with a use case diagram and a sequence diagram. The use case diagram represents the interaction between the actors involved: part designer, fixture designer, manufacturing planner and machine tool operator; and the different tasks to be carried out by the fixture KBE application. The sequence diagram shows the behaviour of the scenario, the tasks and the data that are passed among them [10].

The Definition of Top-Level Functions aimed the creation of a set of high-level software function templates independently of any KBE developing environment. The implementation of such functions in a KBE application allows generating fixture solutions, which are compliant with the functional requirements and the constraints to be fulfilled. These functions were represented graphically using a notation based on IDEFO [9]. A sequence diagram showing the internal interactions was created for each function. Such diagram represents the inference process.

The Product Knowledge Model, where product refers to fixture, was structured into UoK and it was created following the MOKA methodology for the informal level (ICARE forms) and the formal model was represented using UML. In order to be able to reuse pieces of the model in other developments, the concept of knowledge template was adopted from CommonKADS [11].

Once the independent implementation models are created there is a need to map them into the data structure and functions of the API of the implementation system. For instance, two systems such as ICAD and CATIA v5 have a completely different internal data structure and set of API functions [7]. As a consequence, models specific to the implementation system should be created prior to the application coding.

## 4. Fixture design process and knowledge modelling

Most of the methodologies developed for the design of fixtures and jigs are rather linear, proposing a set of top-level tasks to follow [12–14]. A more detailed approach based on the creation of an IDEFO model is presented by Cecil [15]. In any design process, the reasoning cycle goes forward and backward, from the problem domain to the solution domain. It is an iterative analysis and synthesis loop where requirements are the input and design solutions are the output. A functional approach was adopted in this research for the formalization of the fixture design process, where Fixture Functional Components (FFC) were defined and mapped to Fixture Commercial Elements (FCE) [9].

The adoption of a fixture functional design stage followed by a fixture-detailed design has implications, in particular, in the units of knowledge needed to develop a KBE application for fixture design. Several authors are also focused on the development of expert applications for fixture design, none of them considering the fixture functional design stage, and using different artificial intelligence implementation techniques such as:

- Multi-agent implementation with genetic algorithms and neural networks [3].
- Production rules [6,7].
- Case-based reasoning [4,14].

Because KBE applications need to create geometrical representations of the solutions, a development environment with a

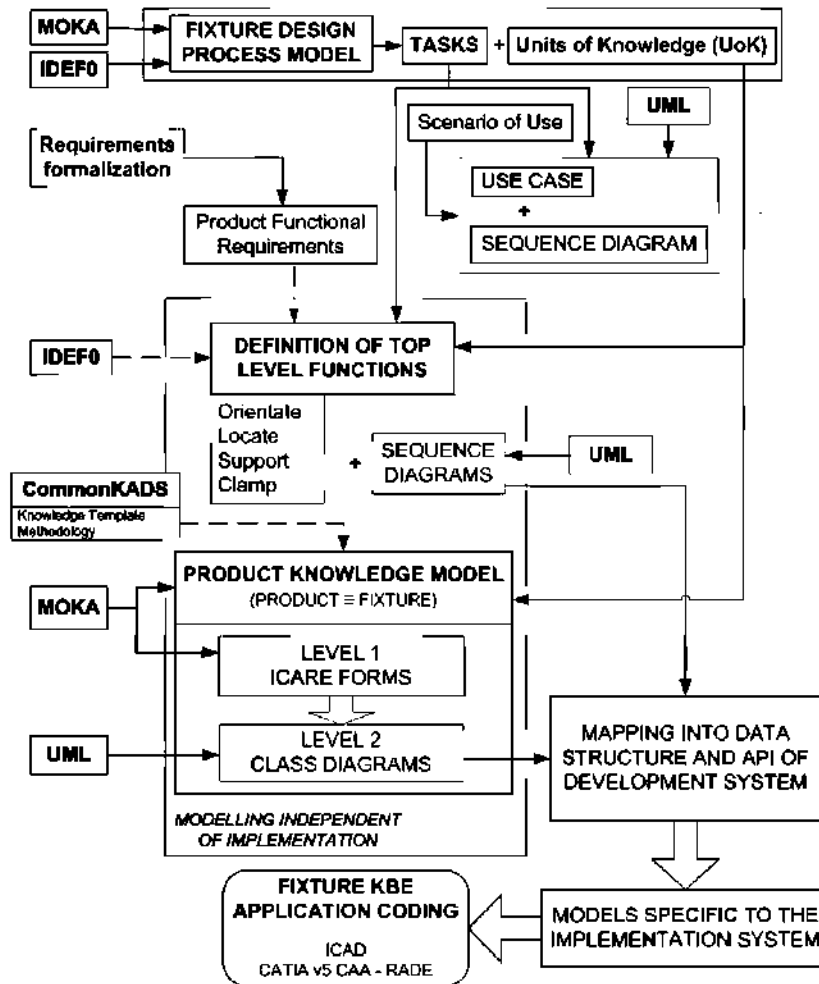


Fig. 1. Methodology proposed to develop a knowledge model applied to fixture design.

geometric kernel is needed. Any geometric kernel has its own internal data structure, and a set of API functions are available in the form of programming libraries. In this context, the development of a KBE fixture design application demands to map a generic and independent knowledge models to the data structure and functions of the development environment. As a consequence, only when the generic models are available then the fixture design process could be implemented in a knowledge-based application.

Considering the first stage of the methodology, Design Process Model (Fig. 1), the activity A3 *Detail Fixture Design Plan* was subdivided into three main sub activities: A31 *Specify Fixture requirements*, A32 *Create Fixture Functional Design* and A33 *Create Fixture-Detailed Design*. A set of Units of Knowledge were identified as required: *Part Geometry*; *Part Geometric Dimensioning and Tolerancing*; *Machining Process and Operations*; *Fixture Functional Requirements*; *Fixture Functional Functions*; *Fixture Functional Design Rules*; *Fixture Functional Elements*; *Fixture-Detailed Design Rules* and *Fixture Commercial Elements* (Fig. 2) [10]. Such UoKs constitute the Fixture Product Knowledge Model.

## 5. Fixture Product Knowledge Model

Following the MOKA approach, this model was divided into two levels: Level 1 based on the use of the ICARE forms and Level 2 based on the creation of UML class diagrams.

In Level 1, the analysis of: input, control, output and mechanism; elements from the IDEF0 (Fixture Design Process Model) led to create a set of ICARE forms. Fig. 3 shows the geometric tolerances entity form, which is part of the UoK *Part GD&T*. The entity form complements the information to the activity A31 *Specify Fixture Requirements*, and it describes in detail the features defined within the *geometric\_tolerance* concept. For each IDEF0 activity a MOKA activity form and a set of entity forms were created [10].

In Level 2, the starting point for each UoK was the set of MOKA entity forms. This level demands to identify and model objects, attributes and relationships that constitute the concepts of the system. The modelling was done in UML class diagrams. This part of the model corresponds to the domain knowledge in CommonKADS [1]. The inference knowledge was represented in the definition of the top-level software functions in sequence diagrams [10].

### 5.1. Unit of knowledge: fixture functional requirements (FFR)

The FFR UoK contains the formalization and definition of the fixture functional requirements. A functional requirement specifies what the fixture has to do. The definition statement is based on a semantic structure that considers the following components: *Action*, *Object*, *Resource* and *Qualifiers*. The *Action* component is defined by an active verb to declare the function that must be satisfied by a fixture design solution, e.g. orientate, support, clamp

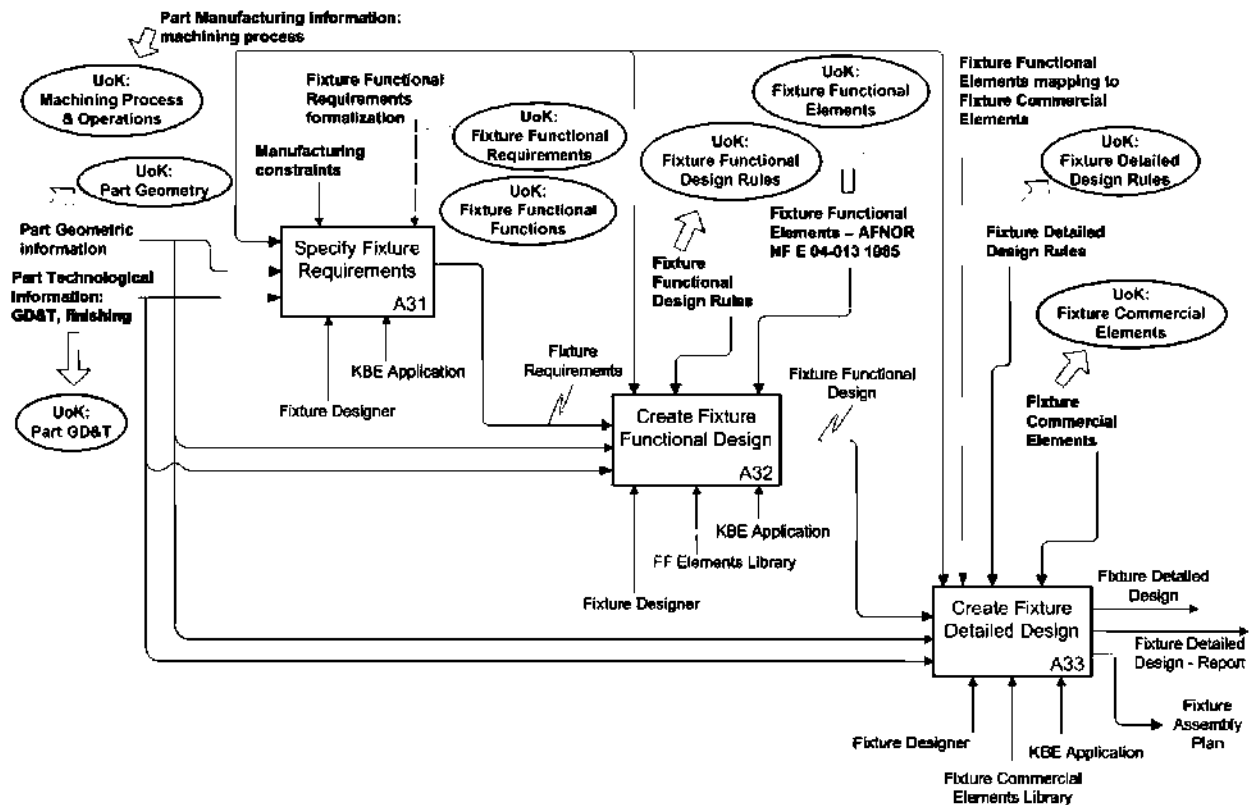


Fig. 2. Activity A3 detailed fixture design plan and the defined Units of Knowledge (UoK).

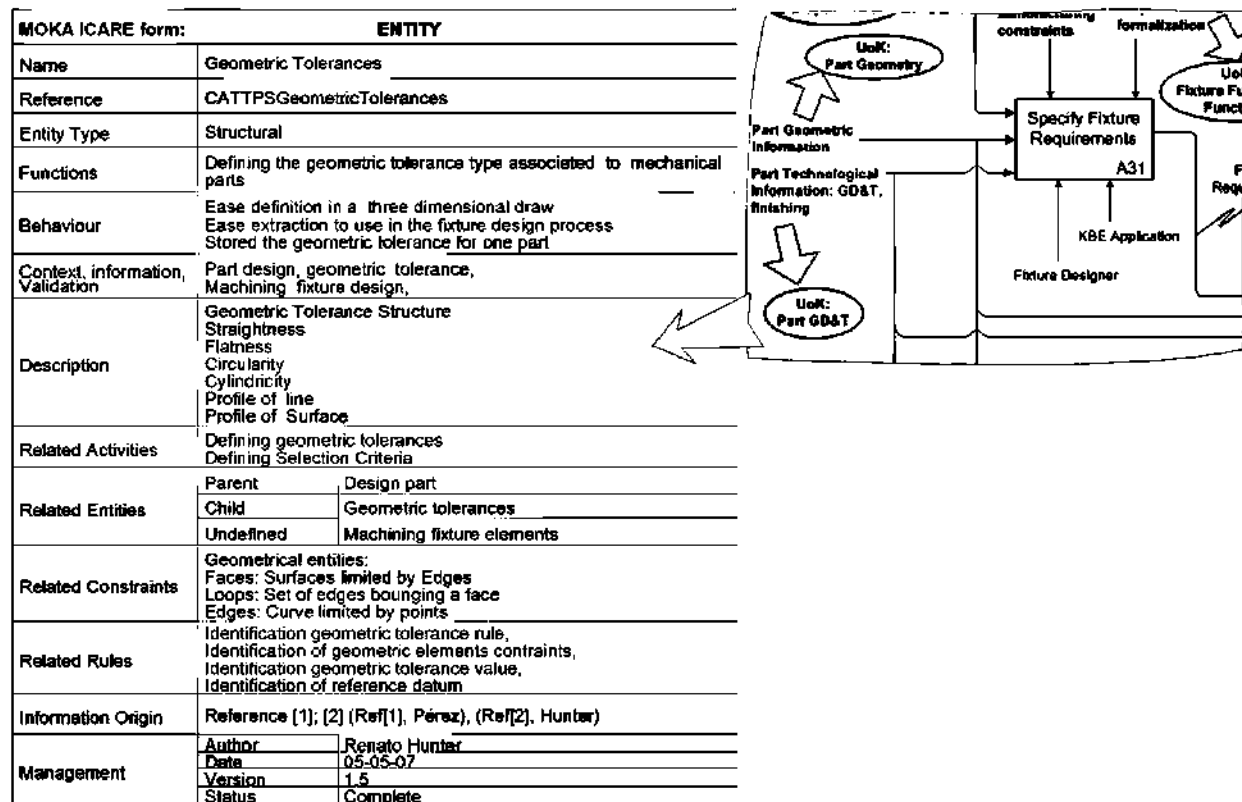


Fig. 3. IDEFO-ICOM elements and ICARE form.

and locate. The *Object* component defines the elements on which the action is executed. The *Resource* component defines where the action will be executed. The *Qualifiers* define the limits of each

functional requirement and represent the constraints, e.g. part tolerances, machining operation, etc. All the modal adverbs are not considered as a *Qualifier*, because they do not have a

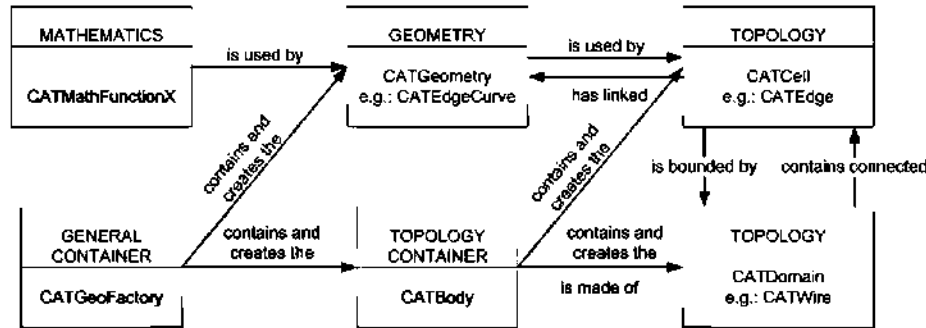


Fig. 4. Top-level CATIA v5 geometric modeller structure.

quantitative value, and as a consequence, they cannot be measured neither validated [9]. In this UoK, the class *functional\_requirement* is an abstract super-type of four sub-classes: *orientation\_requirement*, *clamping\_requirement*, *locating\_requirement* and *supporting\_requirement*. Other defined entities are: *cost\_requirement*, *machine\_tool\_requirement* and *machining\_feature\_requirement* [9,10]. This information is an input to the fixture functional functions UoK.

#### 5.2. Unit of knowledge: fixture design functions (FDF)

The representation of the fixture design function is carried out in two levels. The first level establishes the definition of the fixture functions, inspired by the IDEF0 notation, independent of the development environment used to implement them, e.g.: clamping, supporting, etc. [9]. The second level comprises class and sequence diagrams using UML notation [10].

Each function needs input data from the other units of knowledge: part information (i.e.: part geometry and dimensions and tolerances, material), machining information (i.e.: machining operations, machining strategy, cutting parameters and volumes to remove), functional requirements with constraints (i.e.: working area and base plate size), fixture functional design rules (i.e.: guiding and locating pins, fixing of remnants and fixing of inlets), fixture-detailed design rules (i.e.: distance between fixing screws, vacuum system definition and kind of clamps). Additionally, the definition of the functions demands different analysis models related to the constraints, e.g.: accuracy [16], accessibility [16] and stability [17]; and optimization methods [18,19].

#### 5.3. Unit of knowledge: part geometry (PG)

The automation of fixture design demands to analyse the geometry of the part [7,20], the result from such analysis is used as input for the fixture design functions. If no information about the part geometry is given by the designer interactively then the whole part geometric analysis has to be carried out by the fixture design application. Such analysis has to provide enough information to define properly the location of the different fixture elements. Design features recognition is a research discipline in itself and different algorithms have been proposed based on different approaches: visibility map and convex hull, graph-based B-rep model, volume decomposition, feature templates and discretization techniques [21,22]. The part geometry knowledge unit describes a detailed view of the geometric and topological entities that could be used to define the geometric model, both the part to be machined and of the fixture solution. Ultimately the entities to be considered are determined by the geometric kernel to be used.

The approach adopted in this research was to consider the CATIA v5 geometric modeller and the analysis of the elements contained in the part structure tree using the functions provided in the API. In such geometric modeller, geometry, defined as mathematical functions, is unlimited. Topological concepts are used to set boundaries. The main structure is based on the following concepts: a body is a set of domains (e.g.: lump, shell and wire), a domain contains connected cells (e.g.: volumes connected by faces, faces connected by edges and edges connected by vertices), cells are bounded by domains of lower dimension (e.g.: shell, loop and vertex) and cells have associated geometry (e.g.: CATSurface, CATEdgeCurve, CATMacroPoint) (Fig. 4).

#### 5.4. Unit of knowledge: geometric dimensioning and tolerancing (GD&T)

Because of its direct link to the geometric definition, this UoK was adopted from the technological product specification (TPS) domain defined in CATIA v5. The semantic structure of this UoK contains seven super classes or types to define the different types of tolerances: *CATITPSForm*, *CATITPSDimension*, *CATITPSOrientation*, *CATITPSPosition*, *CATITPSRunOut*, *CATITPSDatum*, *CATITPSRoughness* and *CATITPSNonSemantic*.

#### 5.5. Unit of knowledge: machining process (MP)

A machining process is composed of a sequence of machining operations. The information associated with a machining operation can be divided into two groups: geometric information and technological information. The geometric information defines the volumes to be removed from the raw part, including the volume swept by the cutting tool during the approaching and retracting movements and their position regarding a coordinate reference system. The technological information defines the way how the machining will be performed: machining strategy, cutting tool and cutting conditions. Using both groups of information and appropriated algorithms, the tool path associated to the operation can be calculated. From the perspective of fixture design, the operation geometric information defines the volumes the fixture should be kept away to avoid interferences. It also allows identifying remnants that should be clamped to avoid vibrations during the cutting process. The technological information allows estimating the direction of application of the cutting force and its value. This information is essential for a numerical evaluation of the clamping force, rigidity and stability of the set-up. The model developed by Rios [23] for drilling and milling operations was taken as a starting point for this UoK [10]. A further analysis should be conducted to evaluate the new standard for numerical control programming ISO 14649 [24,25]. In any case, considering that the machining operations would be defined using a particular

CAM module, then its data structure should be considered to achieve an integrated development.

### 5.6. Units of knowledge: fixture functional elements and fixture commercial elements

This UoK deals with the definition of the data structure needed to create two different catalogues, one for fixture functional elements (FFE) and a second one for fixture commercial elements (FCE). The FFE is defined based on an AFNOR standard [26]. The entity *fixture\_functional\_element* has four attributes: *kind\_of\_technology* (*fixed support*, *fixed locating*, *adjustable clamping* and *adjustable concentric clamping*), *state\_of\_the\_port\_surface* (*rough*, *machined* and *finished*), *function\_of\_the\_technology\_element* (*locate-centre*, *clamp*, *support* and *orientate*) and *kind\_of\_contact\_port\_fixture\_element* (*point* and *surface*) [10].

The definition of the fixture commercial element model is based on fixture modular elements. Fig. 5 shows the top-level classes of this model and the sub-classes of *clamp\_elements*.

The correspondence between the FFEs and the FCEs has a multiplicity of one to many. There are also several manufacturers of FCEs, which makes it more complex. For the implementation of this research only FCEs from a specific fixture manufacturer were considered (Fig. 6).

The development of a standard method to create libraries of components for FCEs was out of the scope of this research. However, to address this aspect two standards should be considered: ISO 10303 [27] and ISO 13584 [28].

### 5.7. Unit of knowledge: functional fixture design rules (FFDR)

This UoK comprises the rules that allow defining a fixture functional solution for machining a part. The rules defined to develop a machining fixture solution are used as input to the fixture functional functions (Section 5.2). Rules are also grouped according to the functional functions: *locate*, *support*, *orientate* and *clamp*; plus *centring* and *interference* (Fig. 7).

One of the main rules of this UoK is the 3\_2\_1 locating principle [29,30]. This principle is based on the definition of six positions with its corresponding orientation vector to constrain the six degrees of freedom of the part. As a consequence of its application, the workpiece position and orientation is uniquely determined. The rule sequence is presented in Fig. 8.

In addition to such basic locating principle a set of rules are needed to create a functional solution that is suitable from the machining perspective. For instance, considering a stability rule the use of three single locating elements in the primary locating face is done only if the shape of the face is triangular. When the

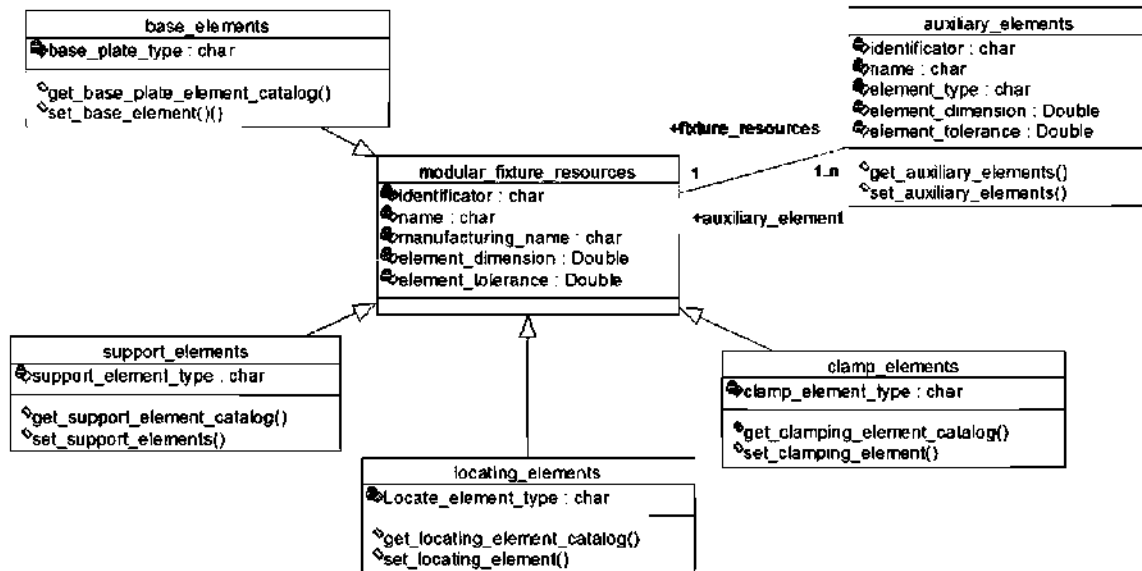


Fig. 5. Sample of the Fixture Commercial Element model.

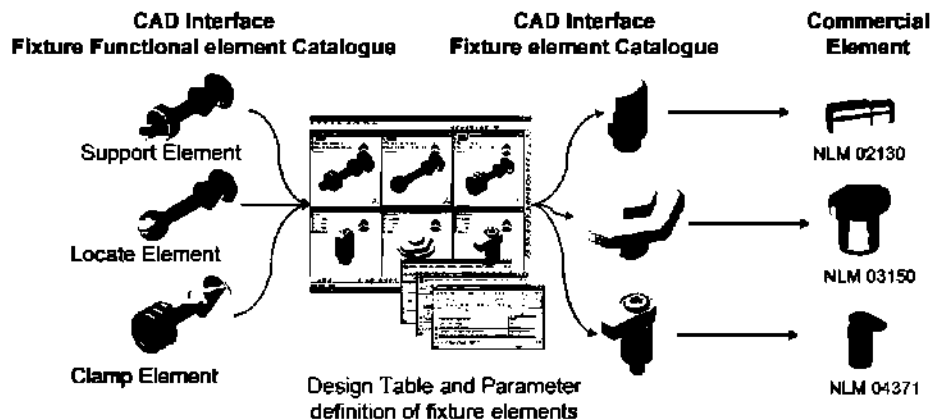


Fig. 6. Sample of the correspondence between FFEs and FCEs.

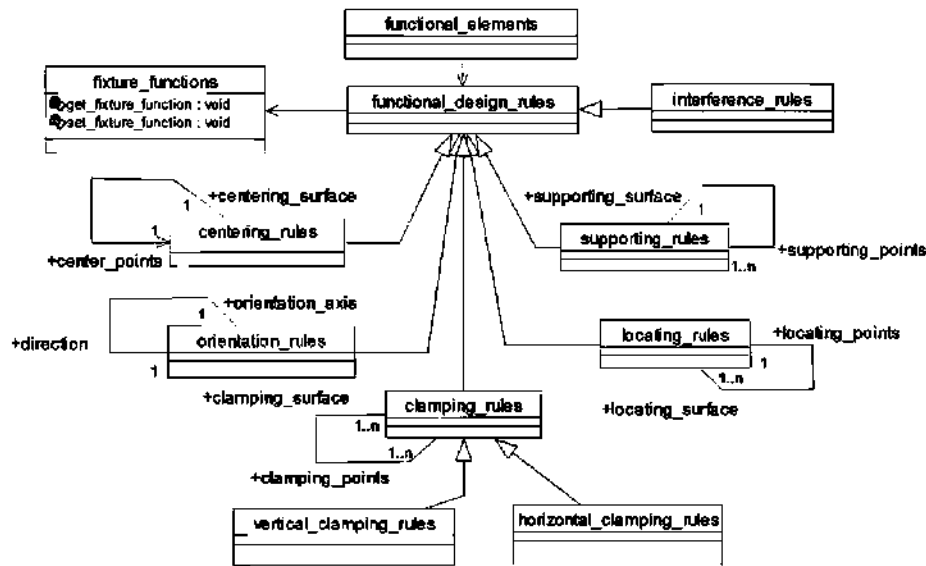


Fig. 7. Top-level functional fixture design rules UoK.

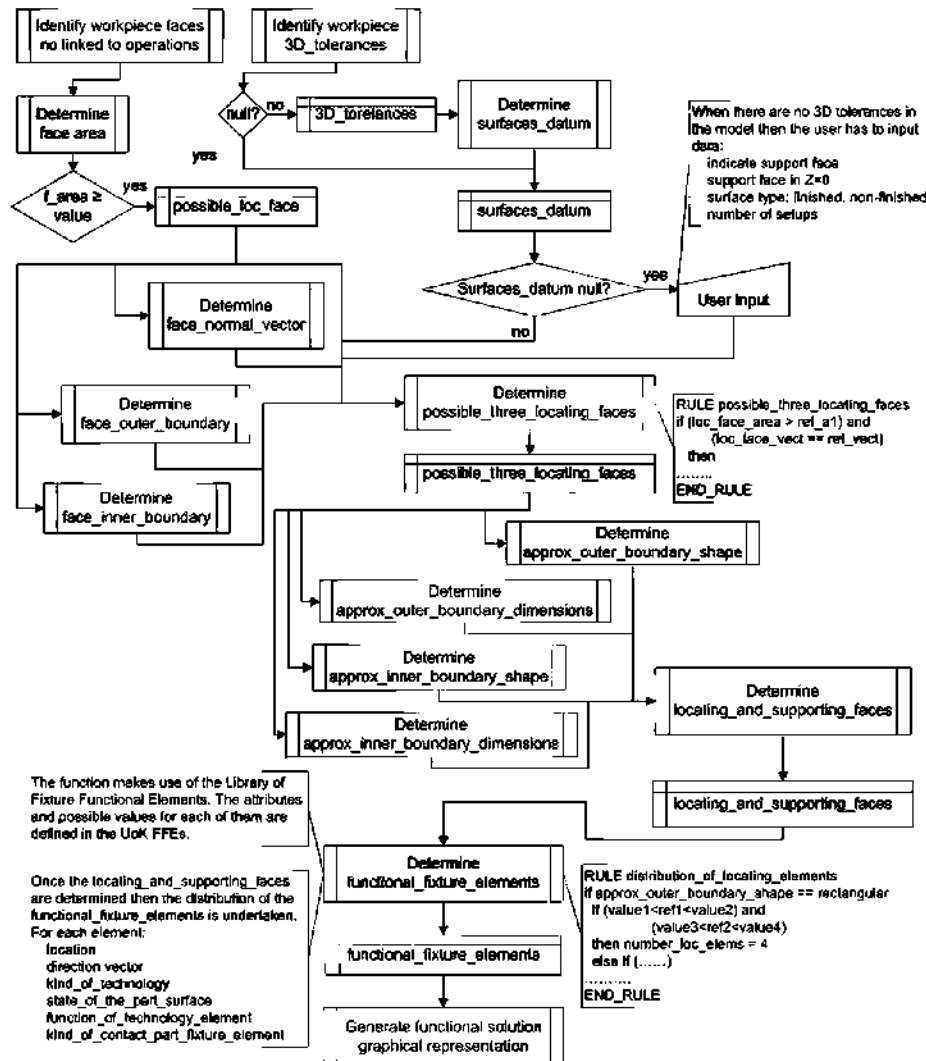


Fig. 8. Sequence of the 3.2.1 locating rule.

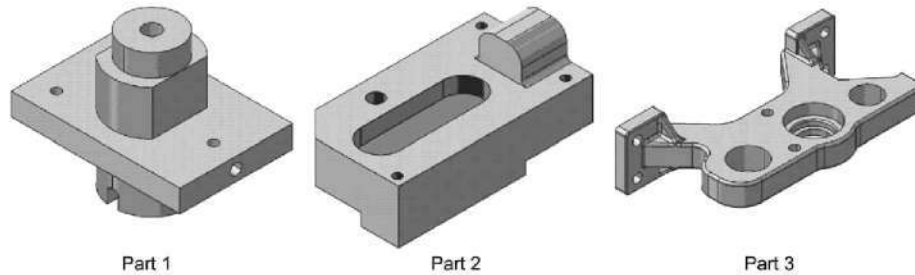


Fig. 9. Parts used in the prototype application validation.

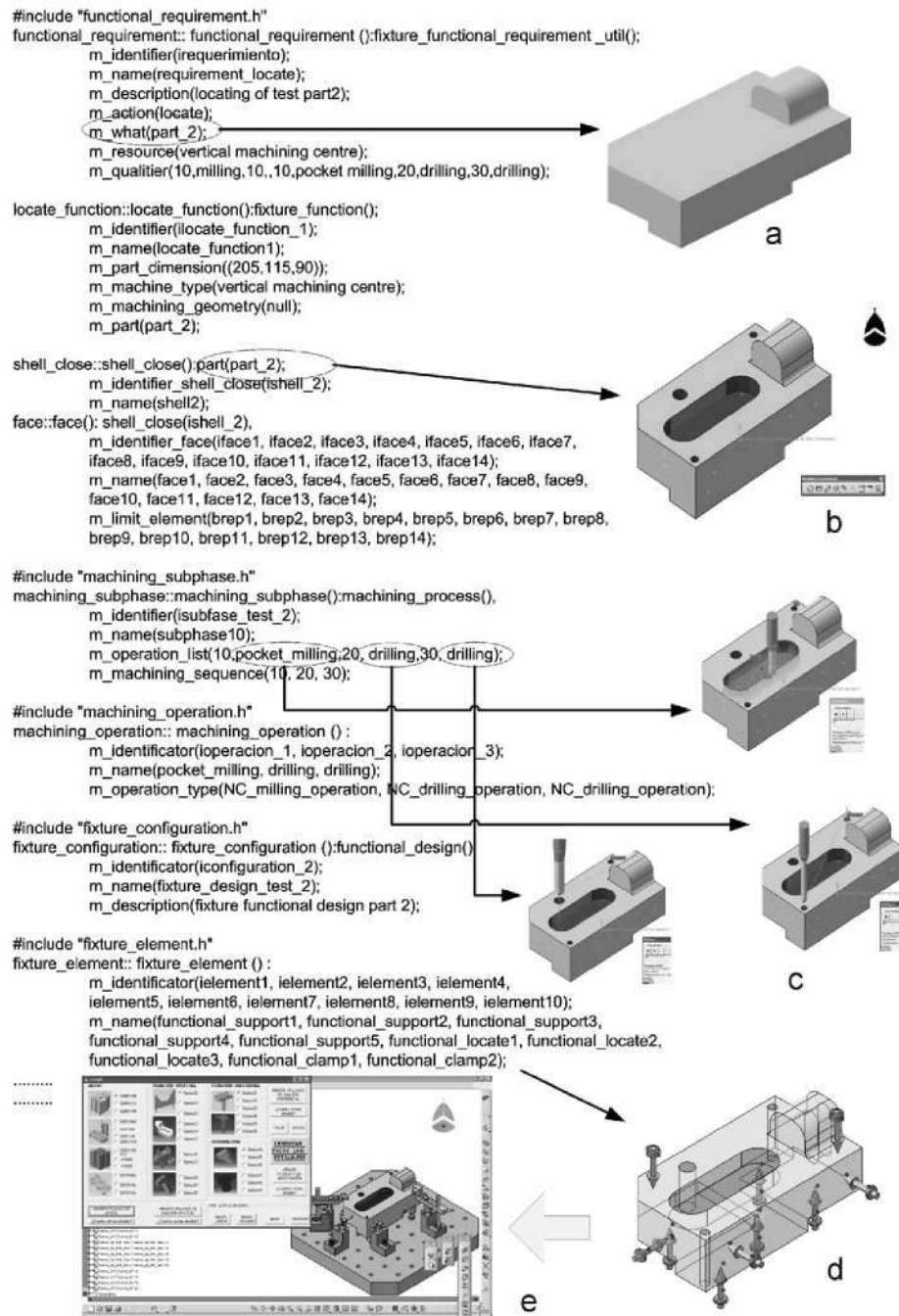


Fig. 10. Fixture design solution example.



shape is rectangular the number of elements is four. However, the final number of elements is determined depending on the size of the primary support face. When a locating surface is also used as support surface then the function of the elements will change from locate to support. If the primary locating face has a circular inner boundary then the associated face is analysed to check if a fixed locator could be selected. Then the function of the element would be locate-centre. The analysis of the possible three locating faces, in terms of inner and outer shape and dimensions, determines the distribution of the functional elements. For the clamping principle, a rule related to the existence of remnants determines the location of clamping functional elements.

### 5.8. Unit of knowledge: detailed fixture design rules (DFDR)

The detailed fixture design knowledge unit makes possible to define a fixture solution using the interpretation and correspondence information between functional elements and commercial elements. The correspondence information allows defining, for each fixture functional element, at least one commercial fixture element (Fig. 6). A fixture functional element contains information about its type, point of application and direction vector. This information is used to select a fixture commercial element from the fixture commercial elements library. The selection of the commercial elements includes the definition of their type, shape and size. Such selection is performed by an algorithm that implements different rules regarding: selection of base plate, selection of clamp type, minimum requirements for the distance between elements, size of remnants, number of set-ups for the workpiece, volumes to be removed, etc. Rules are also grouped according to the functional functions: locate, support, orientate and clamp; plus centring and interference.

## 6. Fixture Knowledge Model implementation

### 6.1. KBE development environment

The main development environment used in this research was CATIA v5 Component Application Architecture (CAA)—Rapid Application Development Environment (RADE). The architecture is based on the Microsoft's technology named Distributed Component Object Modeller (DCOM). Systems developed with this technology are divided into independent sub-systems, called components. The Application Programming Interface is available both in C++ and Java. A CAA application is made up of callings to the functions available in the API libraries and contained in different CATIA v5 components. Following the methodology proposed in this research (Fig. 1) is possible to identify the lower level tasks that can be carried out by the available functions available in the CAA-API. RADE can be integrated within Microsoft Visual C++ simplifying the compilation and linkage of function and component libraries.

### 6.2. Prototype KBE application for fixture design

The implementation of the units of knowledge that conform the research was done partially in a prototype application. The implementation of the machining process UoK (Section 5.5) is partially pending. This has an effect in two issues: automatic clamping force calculation and automatic interference analysis. Three different parts were selected to test and validate the prototype application and the underlying knowledge model (Fig. 9) [10]. The part number two is taken as example to show the fixture design process (Fig. 10).

The prototype KBE application makes possible the selection of supporting, locating and clamping positions using the initial information provided by the geometric modeller and the machining module. The geometric modeller is used to create the geometric and tolerance information, while the machining module is used to create the machining operations.

The initial information regarding the part is presented in Table 1: material, raw material dimensions and shape, batch size, machine tool to use and fixture type to use. Table 2 shows the technological information of the machining operations.

The KBE application has a predefined group of functional fixture requirements: orient, support, locate and clamp. The qualifiers of such requirements are entered by the user.

Fig. 10 shows the definition of the 'locate functional requirement' taking into account the initial conditions defined in Table 1 and Table 2. The displayed code provides a partial view of the instantiation of the fixture design knowledge model. Fig. 10b presents the material to be removed from the part, providing the geometry for the machining operations. The code with the associated geometric entities is shown next to it. Fig. 10c shows the operations defined with CATIA v5 Prismatic Machining. Based on the sequence presented in Fig. 8 (3\_2\_1 locating rule), the prototype application creates the points of application of the fixture functional elements and defines the direction vector. In this case, the solution includes five support points, three locating points and two clamping points. The procedure used by the application includes the identification of supporting, locating and clamping faces through a geometric recognition algorithm. This algorithm has a group of rules that identify the non-machining faces and discards the faces that are associated with a machining operation defined interactively with the Prismatic Machining module. In the next phase, the application uses the information of the supporting, locating and clamping points and vectors to select the appropriated fixture.

The last phase of the fixture design process is to provide the definition of the detailed fixture solution. In this case, the KBE application makes the correspondence between the functional elements and at least one commercial element from the library of commercial fixture elements. A correspondence algorithm is used to define a commercial element based on the information

**Table 1**  
Part 2 initial process conditions.

Specification	Part specification definition
Material	AISI 4130
Initial dimensions (mm)	205 × 115 × 90
Batch size (units)	250
Machine tool	Vertical machining center
Fixture type	Modular fixture

**Table 2**  
Part 2 machining operation information.

Phase	Sub-phase	Operation	Description	Vc (m/min)	Vf (mm/min)	Tool (Ref. Sandvik)
10	10		Milling			
		10	Pocket milling Ø 20 mm	100	200	R216.32- 20030-AC32P
		20	Drilling Ø 9 mm	85	370	R842-0900- 50-A1A 1210
		30	Drilling Ø 16 mm	90	350	R842-1600- 30-A1A 1210

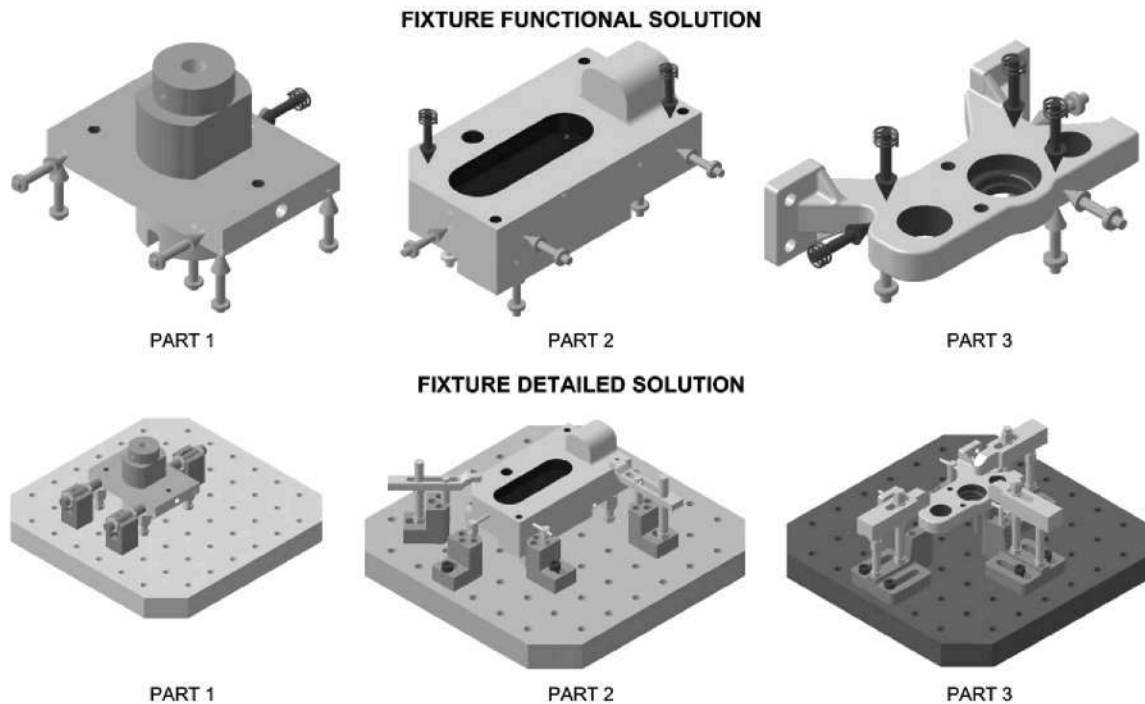


Fig. 11. Functional and detailed fixture solution of the tested parts.

provided by the functional element: kind of technology, state of the part surface, function of the technological element and kind of contact between part surface and fixture element. The selection of the correct element is achieved using a set of IF-THEN rules. Then, the KBE application inserts the commercial element selected as additional components in the model. The information provided by the functional fixture configuration i.e. the coordinates and orientation are used to locate and orientate the element (Fig. 10e). Fig. 11 shows the functional and the detailed fixture solutions of the tested parts [10].

## 7. Conclusions

An approach to model and automate the machining fixture design process has been proposed, implemented and tested. The proposed method complements, especially, the MOKA methodology. In particular, the notation proposed to define Top-Level Functions was evaluated as useful by engineers who are not familiar with software modelling and programming techniques, but have to communicate their requirements to software developers.

As a consequence of using a semi-formalized methodology, the benefits obtained from the development of a KBE application go beyond the use of the application itself. The rationalization, systematization and documentation of the design process allow its improvement and on doing so, to capture tacit knowledge that in regular circumstances is neither formally nor thoughtfully documented.

The creation of a fixture functional solution and representation was positively evaluated. The relevance of such representation resides in three main aspects:

1. It provides a solution independent of any commercial fixture element that can be used to evaluate the fixture concept proposed for the machining of a particular part.

2. It can be used to exchange fixture design information backward and forward between the people involved in the definition and execution of the part manufacturing process prior to the definition of the detailed solution.
3. Because the amount of functional fixture elements is clearly less than the number of commercial elements the generation and implementation of a conceptual fixture solution demands less development effort.

Finally, three main issues demand further research: the complete implementation of the machining process model; the implementation of fixture commercial catalogues following the approach of the standard ISO 13584 [28]; and the implementation of algorithms to evaluate constraints imposed to the fixture solution: accessibility, interference and cost.

## Acknowledgement

The authors would like to express their gratitude to D. Contreras, A. Camarillo, J.V. Jiménez and the personnel from MTA Division EADS-CASA that kindly contributed to this research.

## References

- [1] Schreiber G, Akkermans H, et al. Knowledge engineering and management: the CommonKADS methodology. 3rd ed. London: The MIT Press; 2002.
- [2] Stokes M. Managing engineering knowledge: MOKA—methodology for knowledge based engineering applications. New York: ASME Press; 2001.
- [3] Subramaniam V, Senthil A, Seow K. A multi-agent approach to fixture design. *J. Intelligent Mfg.* 2001;12:31–42.
- [4] Li W, Li P, Rong Y. Case-based agile fixture design. *J. Mater Process Technol* 2002;128:7–18.
- [5] Mervyn F, Senthil Kumar A, Bok SH, Nee AYC. Development of an internet-enabled interactive fixture design system. *Comput Aided Des* 2003;35:945–57.
- [6] Cecil J. TAMIL: an integrated fixture design system for prismatic parts. *Intl J Comput Integrated Manuf* 2004;17:421–34.

- [7] Rios J, Jimenez JV, et al. KBE application for the design and manufacture of HSM fixtures. *Acta Polytech—J Adv Eng—Czech Tech Univ* 2005;45:17–24.
- [8] Mervyn F, Senthil Kumar A, Nee AYC. Fixture design information support for integrated design and manufacturing. *Intl J Prod Res* 2006;44:2205–19.
- [9] Hunter R, Rios J, Perez J, Vizán A. A functional approach for the formalization of the fixture design process. *Intl J Mach Tools Manuf* 2006;46:683–97.
- [10] Hunter R. Definition and integration of requirements and mfg. functions in a KBE system applied to machining fixture design. Ph D Thesis, Polytechnic Univ. Madrid, 2004.
- [11] Hunter R, et al. Knowledge model as an integral way to reuse the knowledge for fixture design process. *J Mater Process Technol* 2005;164:1510–8.
- [12] Henriksen E. Jig and fixture design manual. New York: Industrial Press; 1973.
- [13] Cecil J, Mayer R, Hari U. An integrated methodology for fixture design. *J Intelligent Manuf* 1996;7:95–106.
- [14] Kakish J, Zhang P, Zeid I. Towards the design and development of a knowledge-based universal modular jigs and fixtures system. *J Intelligent Manuf* 2000;11:381–401.
- [15] Cecil J. Computer aided fixture design: using information intensive function models in the development of automated fixture design systems. *J Manuf Systems* 2002;21:58–71.
- [16] Wu Y, Rong Y, Ma W, LeClair SR. Automated modular fixture planning: accuracy, clamping and accessibility analyses. *Robotics CIM* 1998;14:17–26.
- [17] Liao YJG, Hu SJ. Flexible multibody dynamics based fixture workpiece analysis model for fixturing stability. *Intl J Machine Tools Manuf* 2000;40:343–62.
- [18] Pelinescu DM, Wang MY. Multi-objective optimal fixture layout design. *Robotics Comput Integrated Manuf* 2002;18:365–72.
- [19] Kaya N. Machining fixture locating and clamping position optimization using genetic algorithms. *Comput Ind* 2006;57:112–20.
- [20] Wu Y, Rong Y, Ma W, Leclair S. Automated modular fixture planning: geometric analysis. *Robotics Comput Integrated Manuf* 1998;14:1–15.
- [21] Owodunni O, Hinduja S. Evaluation of existing and new feature recognition algorithms—part 1: theory and implementation. *Proc Inst Mech Eng Part B: J Eng Manuf* 2002;216:839–51.
- [22] Rubio MA, Perez JM, Rios J, Vizán A, Marquez JJ. A procedure for plastic parts demoldability analysis. *Robotics Comput Integrated Manuf* 2006;22:81–92.
- [23] Rios J. Integration of CNC mach. tool programming functions by an OO application based on a machining operation information model. Ph D Thesis, Polytechnic Univ. Madrid; 1996.
- [24] ISO. Industrial automation systems and integration. Physical device control. Data model for computerized numerical controllers. Part 10: General process data, ISO 14649-10: 2004.
- [25] ISO. Industrial automation systems and integration. Physical device control. Data model for computerized numerical controllers. Part 11: Process data for milling, ISO 14649-11: 2004.
- [26] AFNOR. NF E 04-013-1985, Dessins techniques, Dessins d'operations, symbolisation des prises de pieces, 1985.
- [27] ISO. Indus. automation systems and integration. Product data representation and exchange. Part 240: Application protocol: Process plan for machined products, ISO 10303-240: 2005.
- [28] ISO. Industrial automation systems and integration—Parts library—Part 1: Overview and fundamental principles, ISO 13584-1: 2001.
- [29] Rong Y, Zhu Y. *Computer-Aided Fixture Design*. New York: Marcel Dekker; 1999.
- [30] Boerma JR. *The design of fixtures for prismatic parts*. Laboratory of Production Engineering, University of Twente; 1990, ISBN 90-9003483-8.